[예제 7-1] ex07-01.c1

```
#include <unistd.h>
#include <stdlib.h>
#include <sys/types.h>

void cleanupaction(void);

main()
{
        pid_t pid;
        int i;

        for(i = 0; i < 3; i++)
        {
                printf("before fork [%d]\n", i);
                sleep(1);
        }

        pid = fork();

        if(pid > 0)
        {
                for( ; i < 7; i++)
                {
                        printf("parent [%d]\n", i);
                        sleep(1);
                }
                atexit(cleanupaction);
        }
        else if(pid == 0)
        {
                for( ; i < 5; i++)
                {
                        printf("child [%d]\n", i);
                        sleep(1);
                        execl("/bin/ls", "ls", "-l", (char *)0);
                }
        }
        else
```

```
        {
                printf("fail to fork child process\n");
        }

        exit(0);
}

void cleanupaction(void)
{
        printf("clean-up-action\n");
}
```

[예제 7-2] ex07-02.c

```
#include <unistd.h>
#include <sys/types.h>

main()
{
    pid_t pid;
    int i = 0;

    i++;
    printf("before calling fork(%d)\n", i);

    pid = fork();    /* fork 호출이 성공하면 자식 프로세스가 생성된다. */

    if(pid == 0)
        /* 자식 프로세스가 수행할 부분 */
        printf("child process(%d)\n", ++i);
    else if(pid > 0)
        /* 부모 프로세스가 수행할 부분 */
        printf("parent process(%d)\n", --i);
    else
        /* fork 호출이 실패할 경우 수행할 부분 */
        printf("fail to fork\n");

}
```

[예제 7-3] ex07-03.c

```
#include <unistd.h>

main()
{
    printf("before executing ls -l\n");
    execl("/bin/ls", "ls", "-l", (char *)0);
    printf("after executing ls -l\n");
}
```

[예제 7-4] ex07-04.c

```
#include <stdio.h>

main()
{
    char *arg[] = {"ls", "-l", (char *)0};
    printf("before executing ls -l\n");
    execv("/bin/ls", arg);
    printf("after executing ls -l\n");
}
```

[예제 7-5] ex07-05.c

```
#include <unistd.h>

main(int argc, char *argv[])
{
    int i;
    for(i = 0; i < argc; i++)
        printf("[%d]%s\n", i, argv[i]);
}
```

[예제 7-6] ex07-06.c

```
#include <unistd.h>

main()
{
    execl("ex07-05", "apple", "option", (char *)0);
}
```

[예제 7-7] ex07-07.c

```
#include <unistd.h>
#include <sys/types.h>

main()
{
    pid_t pid;

    printf("hello!\n");

    pid = fork();

    if(pid > 0)
    {
        /* parent process */
        printf("parent\n");
        sleep(1);
    }
    else if(pid == 0)
    {
        /* child process */
        printf("child\n");
        execl("/bin/ls", "ls", "-l", (char *)0);
        printf("fail to execute ls\n");
    }
    else
        printf("parent : fail to fork\n");

    printf("bye!\n");
}
```

[예제 7-8] ex07-08.c

```
#include <unistd.h>
#include <stdlib.h>

void func1(void);
void func2(void);

main()
{
    printf("hello!\n");
    atexit(func1);
    atexit(func2);

    printf("bye!\n");
    exit(0);
}

void func1(void)
{
        printf("func1\n");
}

void func2(void)
{
        printf("func2\n");
}
```

[예제 7-9] ex07-09.c

```
#include <unistd.h>
#include <stdlib.h>

void func1(void);
void func2(void);

main()
{
    printf("hello!\n");
    atexit(func1);
    atexit(func2);

    printf("bye!\n");
    _exit(0);
}

void func1(void)
{
    printf("func1\n");
}

void func2(void)
{
    printf("func2\n");
}
```